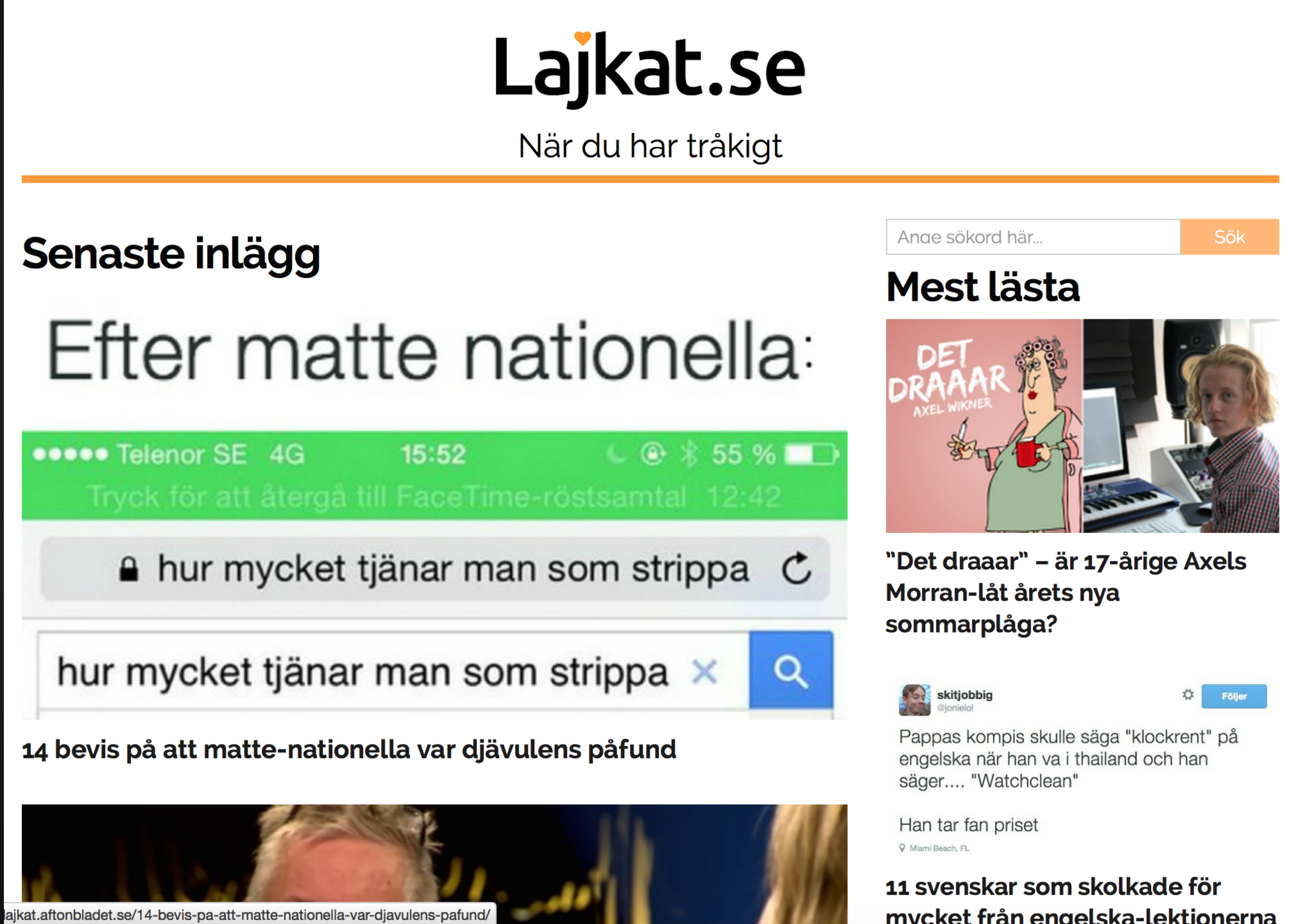
The WordPress Object Cache

Who am I?

Stanislav Khromov
@khromov

AFTONBLADET

SCHIBSTED
MEDIA GROUP



Lajkat.se

När du har tråkigt

Senaste inlägg

Efter matte nationella:
●●●●● Telenor SE 4G 15:52 55 %
Tryck för att återgå till FaceTime-röstsamtal 12:42
hur mycket tjänar man som strippa
hur mycket tjänar man som strippa

14 bevis på att matte-nationella var djävulens påfund

Ange sökord här... Sök

Mest lästa

"Det draaar" – är 17-årige Axels Morran-låt årets nya sommarplåga?

skitjobbig
@jonielol
Följer

Pappas kompis skulle säga "klockrent" på engelska när han va i thailand och han säger.... "Watchclean"

Han tar fan priset

Miami Beach, FL

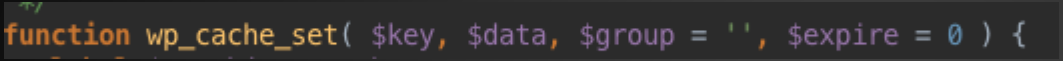
11 svenskar som skolkade för mycket från engelska-lektionerna

lajkat.aftonbladet.se/14-bevis-pa-att-matte-nationella-var-djavulens-pafund/

What is the Object Cache?

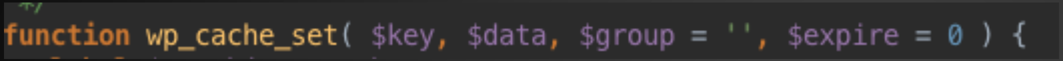
- WordPress only built-in caching layer
- Caches a wide variety of data
- Not reserved for “Objects” in the programming sense.
- Stores different types of content “objects” as posts, users, options & more.
- Non-persistent by default.

Setting a value

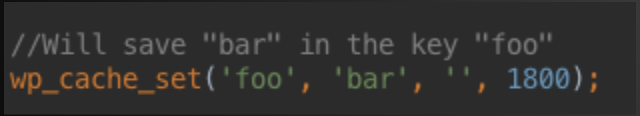


```
function wp_cache_set( $key, $data, $group = '', $expire = 0 ) {
```

Setting a value

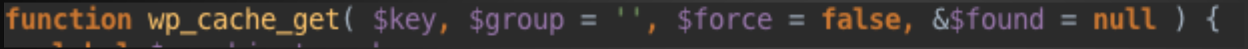


```
*/
function wp_cache_set( $key, $data, $group = '', $expire = 0 ) {
```



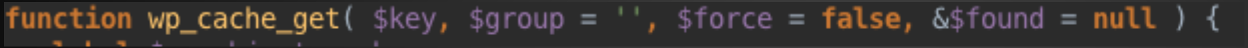
```
//Will save "bar" in the key "foo"
wp_cache_set('foo', 'bar', '', 1800);
```

Getting a value

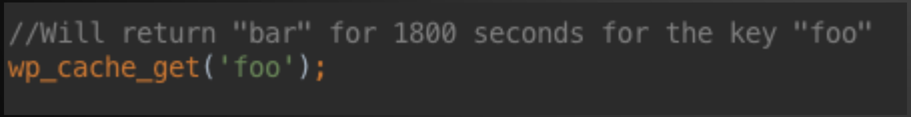


```
function wp_cache_get( $key, $group = '', $force = false, &$found = null ) {
```

Getting a value

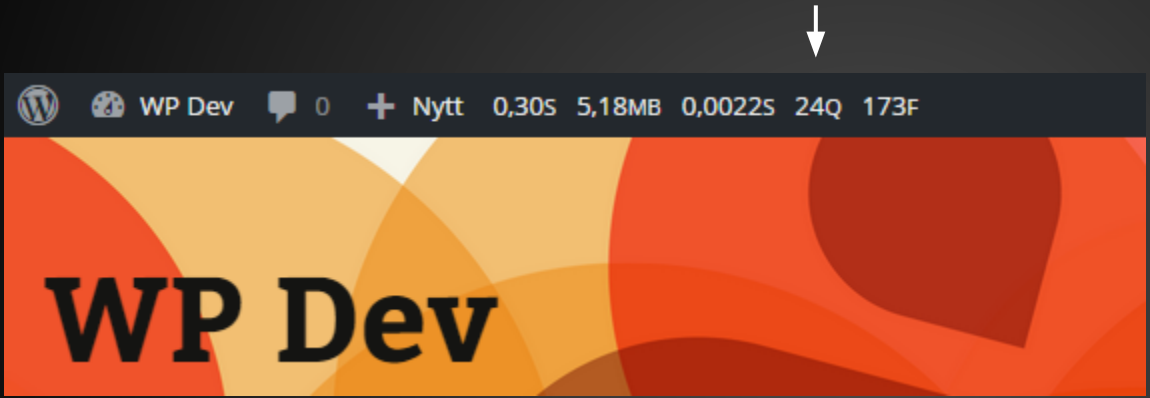


```
function wp_cache_get( $key, $group = '', $force = false, &$found = null ) {
```



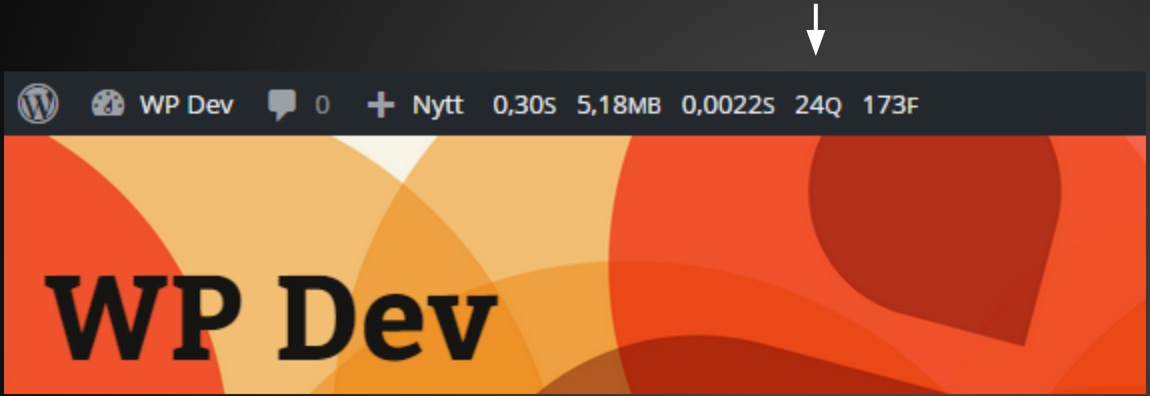
```
//Will return "bar" for 1800 seconds for the key "foo"
wp_cache_get('foo');
```

So what if we disable it?



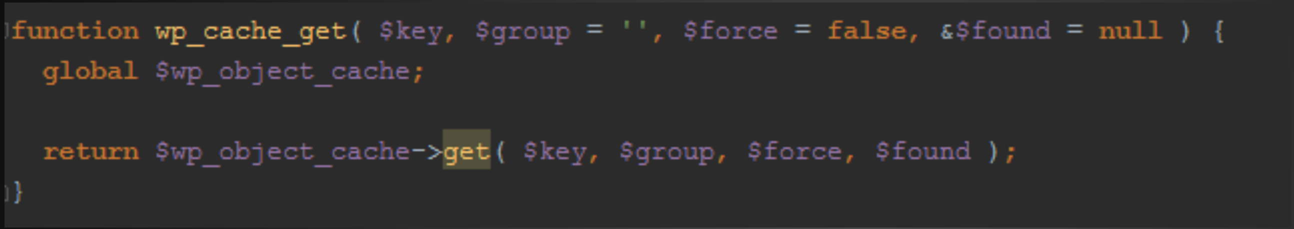
0,30s 5,18MB 0,0022s 24Q 173F

So what if we disable it?

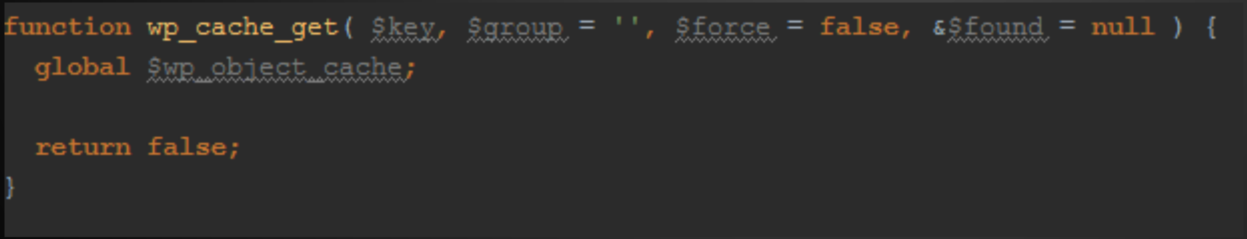


Query Monitor Plugin

So what if we disable it?

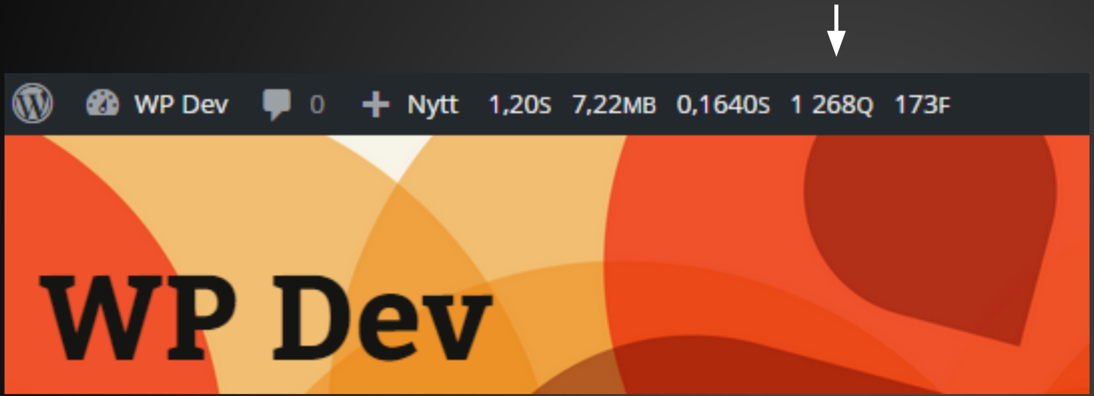


```
function wp_cache_get( $key, $group = '', $force = false, &$found = null ) {
  global $wp_object_cache;

  return $wp_object_cache->get( $key, $group, $force, $found );
}
```

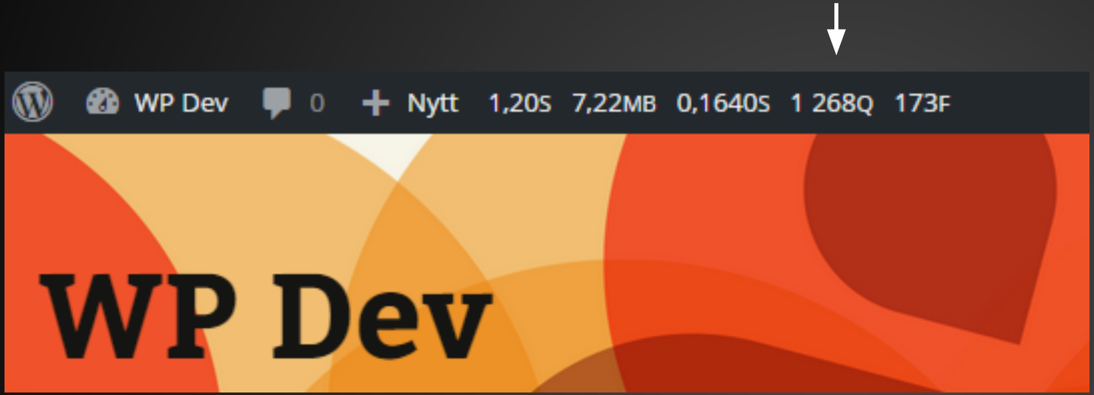


```
function wp_cache_get( $key, $group = '', $force = false, &$found = null ) {
  global $wp_object_cache;

  return false;
}
```

So what if we disable it?



So what if we disable it?



WP Dev 0 Nytt 1,20S 7,22MB 0,1640S 1 268Q 173F

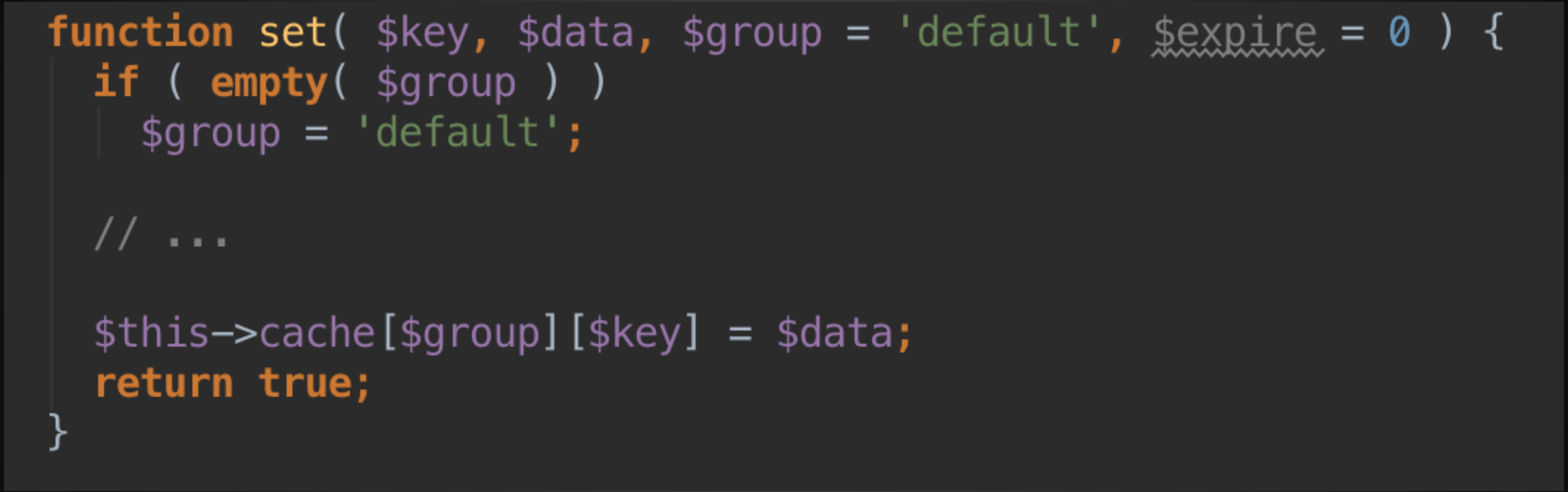
Daaaaamn!

So yeah...

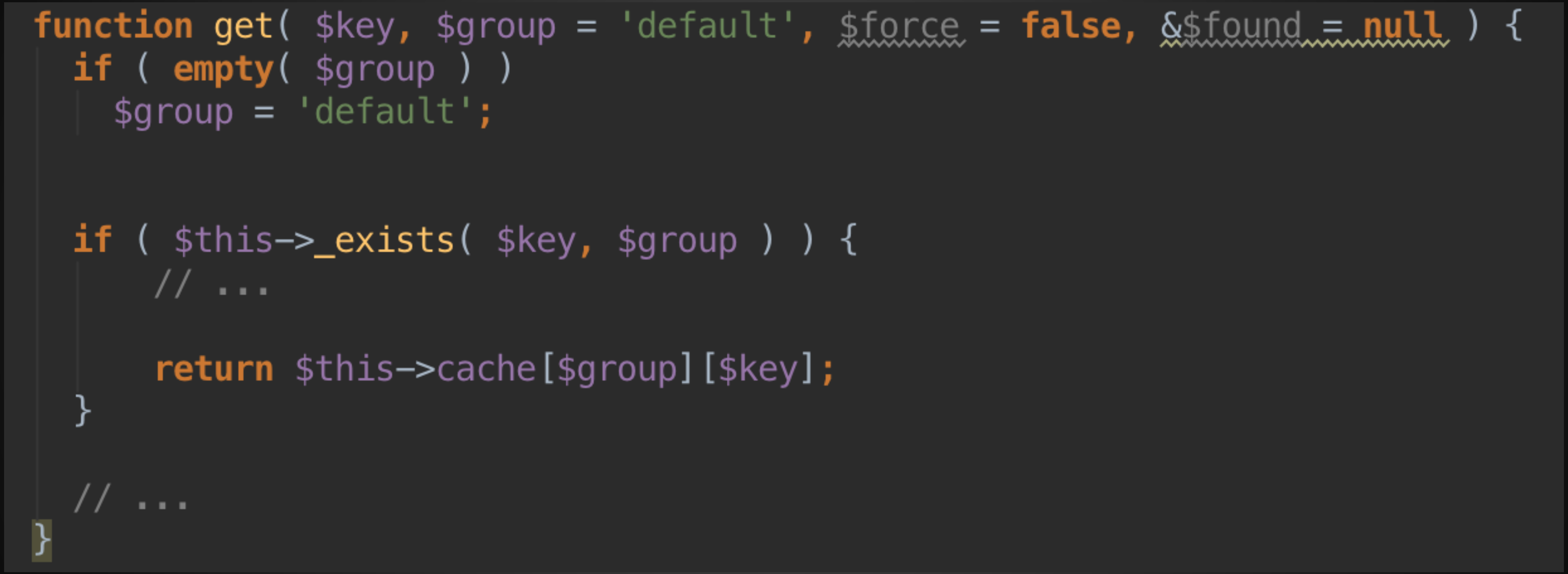
Object Cache is pretty useful!

Non-persistent vs. Persistent cache

- Non-persistent cache works as long as you are in the same request. After the request is served, all data is lost.
  - Implemented in WP as a simple key/value array.



```
function set( $key, $data, $group = 'default', $expire = 0 ) {
  if ( empty( $group ) )
    $group = 'default';

  // ...

  $this->cache[$group][$key] = $data;
  return true;
}
```

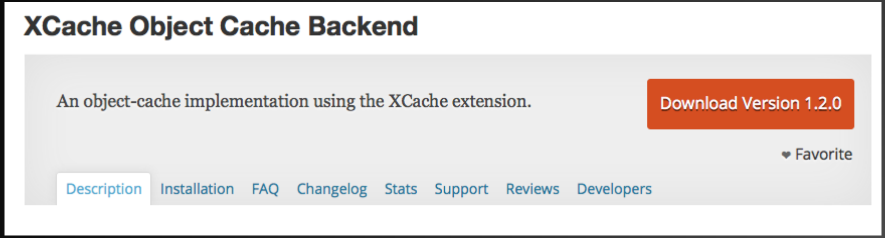


```php
function get( $key, $group = 'default', $force = false, &$found = null ) {
  if ( empty( $group ) )
    $group = 'default';


  if ( $this->_exists( $key, $group ) ) {
      // ...

      return $this->cache[$group][$key];
  }

  // ...
}
```

Non-persistent vs. Persistent cache

- Non-persistent cache works as long as you are in the same request. After the request is served, all data is lost.
  - Implemented in WP as a simple key/value array.
- Persistent cache uses an external storage which is not lost after a request is served.
  - Lots of third party backends available!

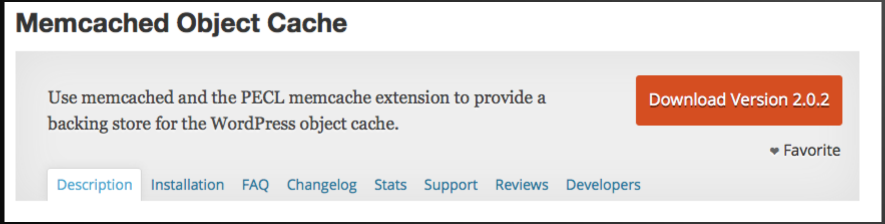
Persistent caches



XCache Object Cache Backend

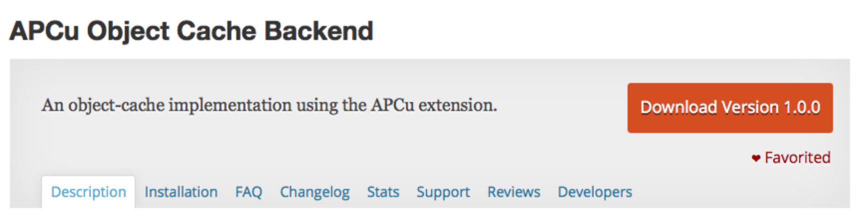
An object-cache implementation using the XCache extension.

Redis Object Cache



Memcached Object Cache

Use memcached and the PECL memcache extension to provide a backing store for the WordPress object cache.



APCu Object Cache Backend

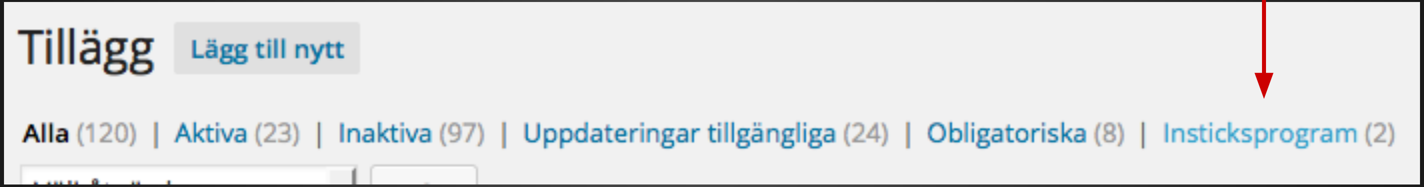
An object-cache implementation using the APCu extension.

Persistent caches

- Rely on a third party mechanism for storage. (Often outside PHP)
- Despite being in plugin repository, not plugins in conventional sense.
- “Drop-ins” - placed into /wp-content/object-cache.php
- Replaces the built-in implementation seamlessly

Persistent caches

“Drop-ins” in English



Tillägg Lägg till nytt

Alla (120) | Aktiva (23) | Inaktiva (97) | Uppdateringar tillgängliga (24) | Obligatoriska (8) | Insticksprogram (2)

Enabling persistent object cache

- In wp-config.php:
  define(‘WP_CACHE’, true);

This will trigger the inclusion of /wp-content/object-cache.php

What does it store?

- Posts & postmeta (custom fields)
- User & usermeta
- Taxonomies
- Options
- Third party plugin data (If they support it)

Issues

- Invalidation is hard. Sane defaults are in core.
- Some third party plugins are improperly implemented.
- May lead to stale data.

Will a persistent Object Cache make my site faster?

Yes

Yes*

Yes*

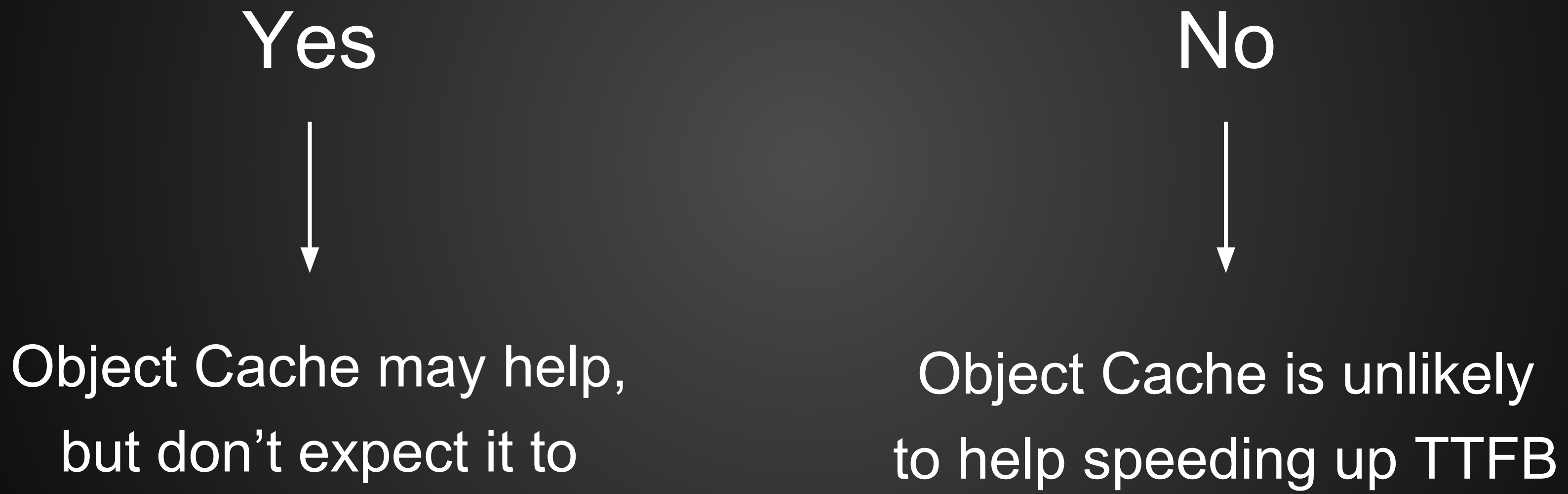
Yes*

Yes*

Speed-up by using Object Cache

- Object Cache works almost exclusively at the database layer
- Content still has to be formatted / rendered
- All filters still run
- MySQL server has similar response time to caching backends under low load

Is your MySQL server overloaded?



Yes

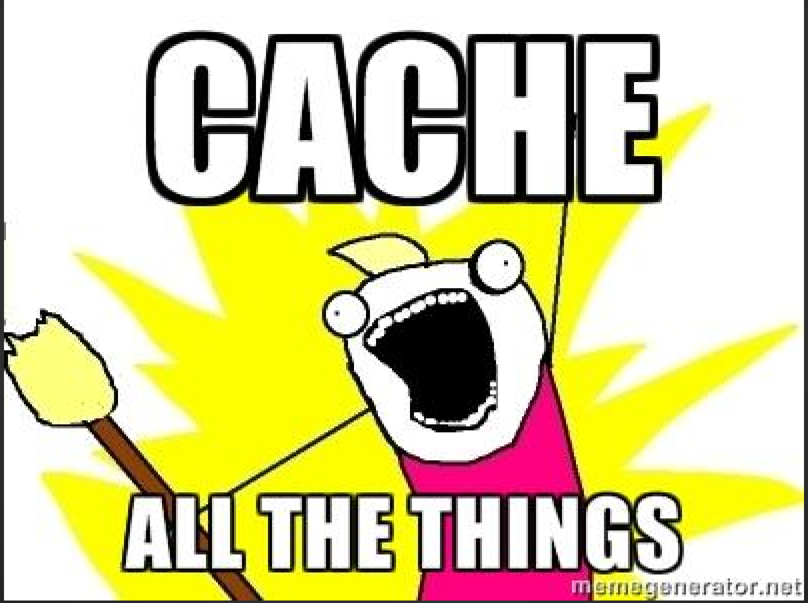
Object Cache may help,
but don’t expect it to

No

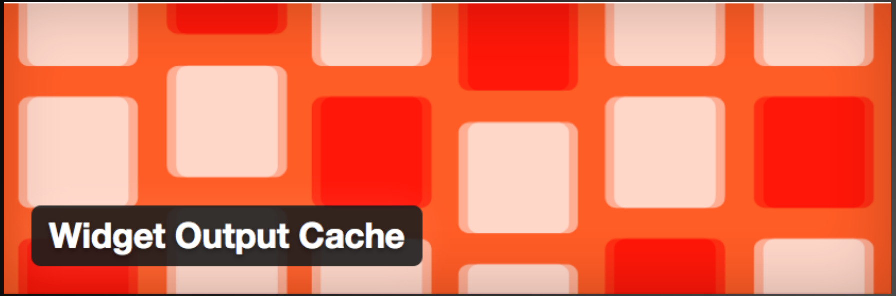
Object Cache is unlikely
to help speeding up TTFB

Getting real gains with the Object Cache

Getting real gains with the Object Cache

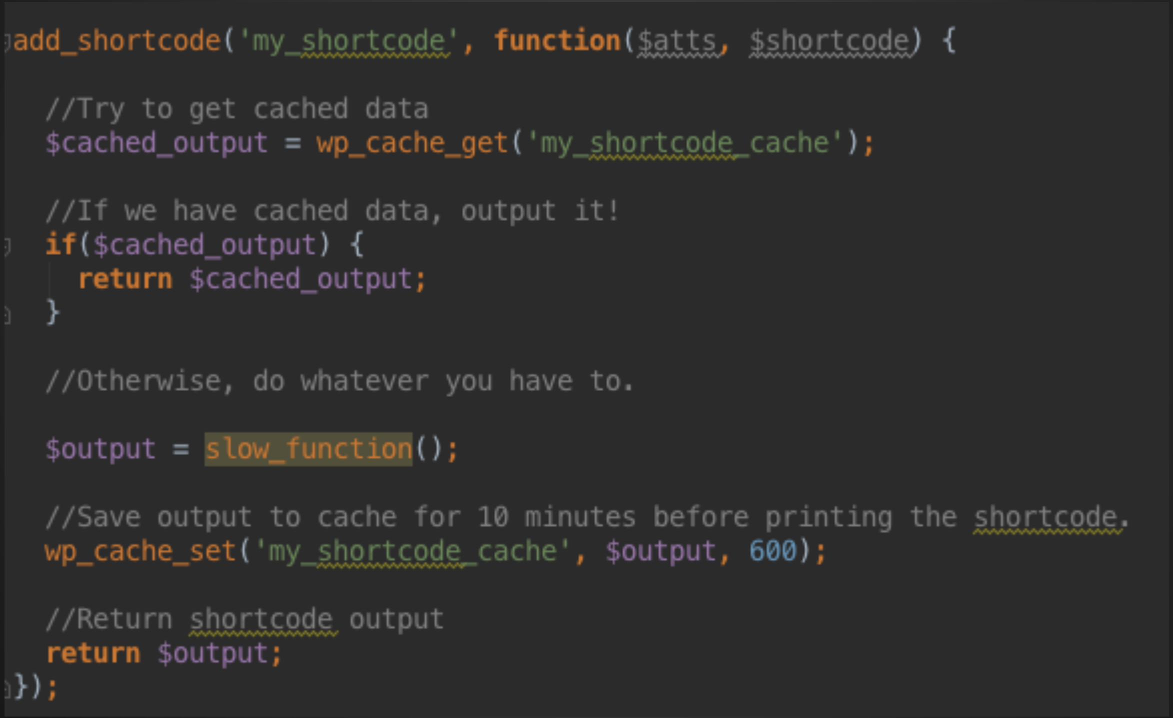


Getting real gains with the Object Cache



- Cache shortcode output!
- Cache anything that is computationally expensive, really…
- Caching is never useless!
- Sky is the limit!

Caching a shortcode



```
add_shortcode('my_shortcode', function($atts, $shortcode) {

  //Try to get cached data
  $cached_output = wp_cache_get('my_shortcode_cache');

  //If we have cached data, output it!
  if($cached_output) {
    return $cached_output;
  }

  //Otherwise, do whatever you have to.

  $output = slow_function();

  //Save output to cache for 10 minutes before printing the shortcode.
  wp_cache_set('my_shortcode_cache', $output, 600);

  //Return shortcode output
  return $output;
});
```

In summary

- Object Cache stops WordPress from doing the same work multiple times
- Non-persistent OC is built-into WordPress
- Persistent cache may help your performance
- Easy to integrate in your own themes and plugins!

Thank you!